



Metodo con retorno

Transcripción

[00:00] Ahora ya tenemos el primero de nuestros comportamientos implementado. Volvemos a nuestra diapositiva y el segundo comportamiento o método que deseamos implementar es el de retirar dinero. Yo ya deposité dinero, y obviamente yo tengo que tener la capacidad de retirar mi dinero del banco para usarlo para X cosas.

[00:23] Volvemos al código, y de la misma forma que nosotros hemos declarado nuestro método depositar, vamos a definir nuestro método retirar. Perfecto. Nuestro paréntesis, y ahora vamos a pensar un poco cómo las veces que tú vas a retirar el dinero del banco, el cajero automático que es lo que necesitas especificar al momento que tú vas a retirar dinero, primero que todo la cantidad de dinero que tú vas a retirar.

[00:58] Entonces, de la misma forma, ponemos double valor y ¿necesitarías especificar la cuenta de la cual estás retirando dinero, sabiendo que estás en tu propia cuenta? Yo creo que no, porque ya hablando ya en objetos, tú estás haciendo referencia a tu cuenta, por lo tanto tú puedes retirar dinero de tu cuenta. Sería muy bueno retirar dinero de cuentas de otras personas, pero no es este caso.

[01:33] Entonces, de igual forma vamos a definir un modificador que va a ser public y digamos que ¿necesitaríamos nosotros saber alguna cosa después de retirar? ¿Necesitaríamos que el método retirar nos retorne alguna información, por ejemplo si el retiro fue exitoso o no? En algunos casos sí. Puede que en otros casos de repente tenga que retornar un mensaje de error o un mensaje

éxito o el valor que estás retirando, o incluso el saldo que te está quedando en la cuenta.

[02:12] En este caso, para hacerlo mucho más simple, yo le voy a decir que quiero que me retorne si fue éxito o fue error. Entonces, eso sería un booleano y lo definimos como el tipo boolean. Recuerden siempre que ustedes tengan situaciones en las que tengan que declarar algo como éxito o error, verdad o falso, es simplemente un boolean, directamente referencia a un boolean.

[02:40] Al igual que el método depositar, vamos a cerrarlo con llaves. Ahora, el método retirar no está dentro del método depositar, si nos damos cuenta a nivel de indentación, los campos y los métodos están en el mismo nivel. Si se dan cuenta aquí, esto está rojo, no los está compilando el código. ¿Por qué? Porque como les dije hay métodos que no retornan valor. Vamos a comentarlo aquí.

[03:16] Y hay métodos que retornan valor. Y en este caso el valor retornado lo especificamos en el tipo de método. Al igual que nosotros especificamos el tipo de variable, aquí cuando especificas el tipo de método, él espera retornar este tipo de dato. Vamos a dejarlo por el momento así y vamos a volver a nuestra prueba de métodos. Venimos aquí.

[03:51] Y ahora, de la misma forma que hemos declarado miCuenta, punto, vemos que ya tengo el método depositar y tenemos el método retirar. Mismo aún sin compilar, él ya nos está anexando con el método retirar. Vamos aquí, y de la misma forma que pasó con depositar, él ya nos está diciendo aquí, bueno, el IDE Eclipse ya nos está sugiriendo, aquí tienes una variable llamada valor que es el tipo double y vamos a ponerle 100. Y punto y coma al final.

[04:28] Vemos que el código compila perfecto y vamos a ir a nuestro método retirar. Si ustedes vieron lo que yo acabé hace ahorita, yo presté la tecla control y con control yo tengo una referencia directa aquí, digamos él me habilita un

link a este método. ¿Por qué lo hice? Porque quizás ya es un poco incómodo estar cambiando entre clase y clase, hacer ese switch rápido.

[04:59] En cambio con control directamente yo puedo ir control saldo y él va a crear un link aquí a saldo, incluso a miCuenta. A la variable miCuenta le doy clic y automáticamente me dice la raíz de esa declaración de variable. Si doy control y selecciono cuenta, me va a llevar a mi entidad cuenta. Control, depositar, me va a llevar a mi método depositar.

[05:29] Ese es uno de los atajos más que Eclipse tiene y que son muy útiles. Ahora, volviendo a nuestro código, vamos a implementar nuestro método retirar. Para yo retirar dinero, yo necesito primero saber si tengo la cantidad de dinero suficiente para poder retirar. Yo no puedo retirar más dinero del que tengo en mi cuenta. Si yo necesito evaluar una condición, entonces necesitamos un if.

[06:00] Ponemos if y empezamos this.saldo, hacemos referencia a m saldo actual. Si mi saldo es mayor o igual que el valor que yo deseo retirar, entonces abrimos llaves y vamos a especificar que a mi saldo, al saldo de esta cuenta, va a ser igual a mi saldo menos el valor que yo deseo retirar. Perfecto. Y si todo sale bien, voy a retornar true. Esa es la palabra reservada de Java para retornar cualquier cosa en realidad.

[06:55] Con return yo especifico que aquí estamos cortando el método y él ya va a retornar este valor y vamos a salir de la ejecución del método. Caso contrario aquí, else, abrimos llaves, return false. Listo. Ya tenemos el método, tenemos un método si se dan cuenta un poco más estructurado. Salvamos, venimos aquí a prueba y vamos a copiar esta línea de aquí.

[07:31] Pegamos aquí abajo. Ya sabemos que hemos depositado 200 y ahora estamos retirando 100. Guardamos, ejecutamos el código y en efecto vemos que después de depositar 200, mi cuenta tiene 500 soles, pero después de retirar 100 soles, mi saldo es 400 soles.

